웹용 영한 기계번역을 위한 문서 전처리기의 설계 및 구현

안동언°, 유홍진, 서진원, 이영우, 정성종
전북대학교 컴퓨터 · 정보통신공학부, 영상정보신기술연구소
여상화, 김태완, 박동인
시스템공학연구소 자연어정보처리연구부

A Preprocessor for English-to-Korean Machine Translation of Web Pages

Dong Un An, Hong Jin Ryu, Jin won Seo, Young Woo Lee, Sung Jong Jeong
Faculty of Computer, Information and Communication Engineering
Chonbuk National University
Sang Hwa Yuh, Tae Wan Kim, Dong In Park
Department of Natural Language Information Processing
Software Engineering Research Institute

요 약

영어 웹 문서를 한국어로 기계번역을 하기 위해서는 HTML 태그를 번역 대상 문장과 분리하는 처리가 필요하다. HTML 태그를 단순히 제거하는 것이 아니라 대상 문장의 기계번역이 종료된 후에 같은 형태의 한국어 웹 문서로 복원하기 위한 방안이 마련되어야 한다.

또한 문서 전처리기에서는 영어 형태소해석기의 성능을 높이기 위하여 번역 단위가 되는 문장의 인식 및 분리, 타이틀의 처리, 나열된 단어의 처리, 하이픈 처리, 고유명사 인식, 특수 문자 처리, 대소문자 정규화, 날짜 인식 등을 처리하여 문서의 정규화를 수행한다.

1. 서론

인터네트 상에서 웹이 매우 빠른 속도로 확산되어 이제는 인터네트를 사용하는 대부분의 사용자들이 웹을 통하여 정보를 전달하고 획득하고 있다. 일반 사용자들이 웹을 통하여 정보를 획득하는데 있어서 걸림돌 중의 하나는 대부분의 웹 문서가 영어로 작성되어 있다는 것이다. 따라서 정보의 신속한 획득을 위해서는 영한 기계번역시스템의 사용이 매우 필요하다.

영어 웹 문서를 한국어로 기계번역을 하기 위해서는 일반 문서와는 다르게 우선 HTML 태그를 번역 대상 문장과 분리하는 처리가 필요하다. 문장만을 번역한다는 목적이라면 HTML 태그를 웹 문서에서 제거하기만 하면 되며 이러한 처리는 매우 간단하다. 그렇지만, 웹 상에서 실시간으로 영어 웹 문서를 번역하여 동시에 웹 브라우저를 통하여 번역된 한국어 웹 문서를 보거나, 영어 웹 문서를 번역한 후에 번역된 한국어 웹 문서를 파일로 저장한 후에 웹 브라우저를 통하여 보기 위해서는 HTML 태그를 단순히 제거하는 것이 아니라 대상 문장의 기계번역이 종료된 후에 같은 형태의 한국어 웹 문서로 복원하기 위한 방안이 마련되어야 한다.

영어 웹 문서의 경우에는 종결 부호의 생략, 강조를 위한 대문자의 잦은 사용, 타이틀이나 단어의 나열 등으로 인하여 형태소 해석에 어려움을 가져온다. 따라서, 문서 전처리기에서는 영어 형태소 분석기의 성능을 향상시키기 위하여 문서의 정규화를 수행한다. 번역 단위가 되는 문장의 인식 및 분리, 타이틀의 처리, 나열된 단어의 처리, 하이픈 처리, 고유명사 인식, 특수 문자

처리, 대소문자의 정규화, 날짜 인식 등을 수행한다.

HTML 태그의 처리와 문서의 정규화를 위한 처리를 lex를 이용하여 구현하였다. lex를 이용함으로써 프로그램의 유지 보수가 매우 용이하고 새로운 현상에 쉽게 대처할 수 있다.

2. 문서 전처리기의 구성

웹용 영한 기계번역을 위한 문서 전처리기의 구성도는 <그림 1>과 같다. 영한 기계번역기는 시스템공학연구소에서 개발하고 있는 "에서로/영한"이다.

본 논문의 문서 전처리기는 [여상화 95]의 시스템을 보완한 것이다. 기존의 시스템은 여러 기능 중에서 문장 분리와 HTML 태그 처리 기능이 미흡하였다. 특히 HTML 태그 처리에 있어서는 단순히 HTML 태그를 분리하는데 그치고 있다. 웹 페이지의 작성에 있어서 웹 페이지 저작 도구를 점차 많이 사용함으로써 웹 문서에 HTML 태그의 수효가 대단히 늘어났고 테이블이나 프레임의 사용이 많아져서 HTML 태그의 처리가 단순하지 않다.

문서 전처리기에서는 하이픈 단어 사전과 고유명사 사전의 두 가지 사전을 사용하고 있다. 하이픈 사전은 복합어에 사용된 하이픈인지 아니면 줄에서 단어가 미처 끝나지 않아서 사용된 하이픈인지 구분하기 위해서 사용한다. 고유명사 사전은 고유명사를 인식하고 대소문자를 정규화하기 위하여 사용한다. 두 가지 사전의 등록어는 MATES/EK[KAIST 92]와 PennTree Bank[Penn 97]에서 수집하였다. 하이픈 단어 사전은 7,200여 개, 고유명사 사전은 7,900여 개의 등록어가 수록되어 있다.

문장, 자질, 태그 분리기는 문서 전처리기의 결과를 영한 기계번역기에서 사용할 수 있도록 변환시켜 주는 인터페이스 역할을 한다. 문서 전처리에서 얻어진 고유명사, 날짜 등의 자질 정보와 HTML 태그 정보를 가진 단어를 문장 단위로 끊어서 영한 기계번역기의 입력으로 보낸다.

웹 문서 복원기는 영한 기계번역이 종료된 후 번역된 단어의 앞뒤에 HTML 태그를 붙여서 완전한 한국어 웹 문서를 만든다.

3. 문서 전처리기의 기능

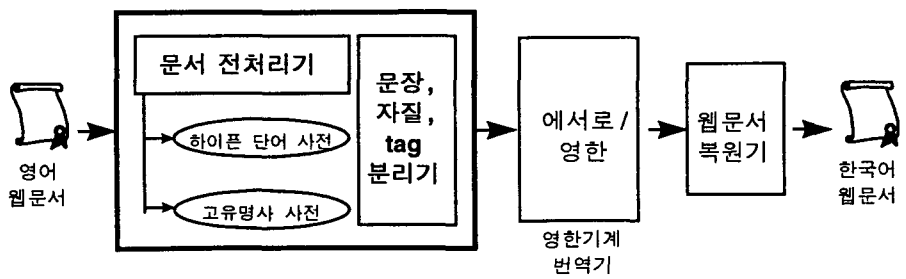
전처리기는 입력 웹 문서에서 나타나는 다양한 현상을 영어 형태소 분석을 하기 전에 미리 처리하여 형태소 분석기의 부담을 덜어주고 성능을 향상시킨다. 또한, HTML 태그를 문장과 분리하여 해당 단어의 자질 정보로 넘긴다.

3.1 문장 분리

영한 기계번역기의 처리 단위는 한 문장이다. 따라서, 웹 문서를 문장 단위로 분리하여야 한다. 웹 문서의 모든 영어 문장에 종결 부호가 있다면 HTML 태그만을 분리한다면 문장을 분리하는데 별다른 문제가 없다. 그러나 웹 문서에는 나열문이 많고 테이블이 많아서 단순히 HTML 태그만을 분리하면 단어들의 나열에 그치게 된다. 타이틀, 나열문, 테이블의 단어 등은 한 문장이 되도록 분리되어야 한다.

(1) 종결 부호: \.|\?|\!|\n
타이틀: </[Tt][Ii][Tt][Ll][Ee]>
테이블 항목: </[Tt][Dd]>
문서 연결하기: </[Aa]>

(1)은 문장 분리의 단서가 되는 문자열을 regular expression으로 표현한 것이다. 종결 부호로 ".", "?", "!"를 이용하며 종결 부호가 없더라도 줄 바꿈이 있으면 "\n"을 단서로 문장을 분리한다. 구나 절로만 구성



<그림 1> 문서 전처리기의 구성도

된 나열문의 문장 분리에 효과가 있다.

테이블의 경우에는 각 항목이 단어들로 되어 있는 경우가 많다. 각 항목들은 따로 번역되어야 한다. 따라서 </TD> 태그를 이용하여 테이블의 각 항목을 문장으로 분리한다. 문서 연결하기 태그로 구성된 연결 문서들의 나열도 독립된 문장으로 분리한다.

3.2 하이픈 처리

하이픈은 복합어에 사용되기도 하고, 라인의 끝에서 단어가 미처 끝나지 않은 경우, 그 단어가 다음 라인에 이어짐을 표시하기도 한다. 라인의 마지막 단어가 하이픈으로 끝나는 경우, 하이픈을 단어의 일부로 보아야 할 지, 아니면 무시해야 할 지를 판단해야 한다. 이 판단을 위하여 하이픈 단어들을 모아 놓은 하이픈 사전을 이용한다. 하이픈 단어 사전의 검색 효율성을 높이기 위하여 "business-as-usual"과 같은 단어는 "business-"와 "business-as-"를 모두 사전에 수록한다. 하이픈 단어 사전에 없으면 하이픈을 제거하고 다음 줄의 단어와 결합한다.

3.3 고유명사 인식

많은 기계번역기에서 미정의어로 인해 번역이 되지 못하는 것은 대부분 고유명사 때문이다. 고유명사를 기존의 사전과 통합하기에는 그 수효가 매우 많다. 따라서, 고유명사를 영한 기계번역을 수행하기 전에 인식하고자 한다. 고유명사 사전에는 남자 이름, 여자 이름, 애완 동물 이름, 조직명, 지명 등이 수록되어 있다. 이 고유명사 사전에 의해 인식된 고유명사에 PROP라는 자질명에 각각 MALE, FEMALE, HUM, PET, ORG, LOC 등의 자질값을 부여하여 영한 기계번역에서 이용할 수 있도록 한다.

(2) 남성: Mr.
여성: Miss, Mrs. Ms.
사람: Dr., Prof, St. Jr.
조직명: Co.

(2)는 인명이나 조직명을 짐작할 수 있는 단서로 사용되는 단어들이다. (3)은 인명을 나타내는 다양한 형태의 regular expression 들이다. 위와 같은 단서들과 규칙들을 이용하여 단어를 인식한 후 고유명사 사전을 검색한다. (3)과 같은 규칙에 해당되지는 않지만 단어의 첫 문자가 대문자이면 고유명사인지를 살펴본다.

```
(3) LAST_NAME: [A-Z][a-z]+|Mc[A-Z][a-z]+|
                    O'[A-Z][a-z]+|La[A-Z][a-z]+
          /* McGovern, O'Kicki, LaFlace */
    [A-Z][a-z]+[ ][A-Z]\.[ ]?{LAST_NAME}
          /* John F. Kennedy */
    [A-Z][a-z]+\.[ ]?[A-Z]\.[ ]?[A-Z]\.
          /* Aho, A. V. */
    [A-Z][a-z]+\.[ ]?[A-Z]\.
          /* Bach, E. */
    [A-Z]\.[ ]?[A-Z]\.[ ]?{LAST_NAME}
          /* J. D. Ullman */
    [A-Z]\.[ ]?{LAST_NAME}
          /* K. Jones */
```

3.4 특수 문자 처리

특수 문자가 단어의 일부로 사용되었을 때는 이를 단어로 인식한다. 예를 들어, "I.B.M."과 "AT&T" 등의 '.'과 '&'는 단어의 일부이다. 웹 문서에서 사용하는 Character Entity(예: ø, ø)도 하나의 단어로 인식한다.

```
(4) Symbol
        [\#\$%&\'\(\)\*\+\'\-\/:;=@\[\]\^_{|}~,]
    Character Entity
        \&[A-Za-z]+\;
```

3.5 대소문자 정규화

웹 문서에서는 타이틀이나 강조하고자 하는 단어에 흔히 대문자를 사용한다. 이러한 문서는 대소문자가 정규화된 형태로 변형하여 처리해야만 동일한 단어가 사전에 대소문자로 따로 등록되는 것을 방지할 수 있고 대문자로만 작성된 문서도 처리할 수 있다.

고유명사를 제외한 모든 영어 단어는 소문자로 정규화한다. 이를 위해 고유명사 사전을 이용한다. 고유명사 사전에 없으면 소문자로 정규화한다.

3.6 날짜 인식

여러 단어로 이루어진 날짜 표현을 하나의 단어로 취급하여 해석 모듈의 부담을 줄인다. 날짜의 인식은

인접한 단어들에 대해서만 수행하며 문장 분리와 대소문자 정규화가 수행된 결과에 대해서 인식을 수행한다.

(5) {MONTH_NAME}[ ]{DATE}\,[ ]{YEAR}

3.7 HTML 태그 처리

HTML 태그를 start tag와 end tag로 구분하고 각 단어에 대해 자질값으로 저장하여 영한 기계번역기에서는 불필요하게 분석하지 않도록 한다. start tag는 <"HTML tag">이며 end tag는 </"HTML tag">이다. 각 단어가 자질값으로 앞뒤의 HTML 태그를 가지고 번역되므로 웹 문서 복원기를 통하여 한국어 웹 문서로 복원할 수 있다.

4. 문장, 자질, 태그 분리기

문장, 자질, 태그 분리기는 문서 전처리기의 결과를 영한 기계번역기에서 사용할 수 있도록 변환시켜 주는 인터페이스 역할을 한다. 문서 전처리에서 얻어진 고유명사, 날짜 등의 자질 정보와 HTML 태그 정보를 가진 단어를 문장 단위로 끊어서 영한 기계번역기로 보낸다. 이를 위한 자료구조는 <그림 2>와 같다.

```
typedef struct f_tag{
        char *tag;
        struct f_tag *next;
} start_tag;

typedef struct b_tag{
        char *tag;
        struct b_tag *next;
} end_tag;

typedef struct list_node{
        char *word;
        start_tag *start_tag_head,
                  *start_tag_tail;
        end_tag *end_tag_head,
                *end_tag_tail;
        char *f_name;
        char *f_value;
        struct list_node *next;
} node;
```

<그림 2> 문장, 자질, 태그 분리 후의 자료구조

문장 내의 단어(char *word)는 전처리기에서 분리한 한 단어나 고유명사, 날짜 등으로 인식된 복합 단위를 저장하는데 사용한다. 영한 기계번역기에서는 이 단어만을 번역 대상으로 처리한다.

HTML 태그는 start tag와 end tag로 구분하여 저장한다. start tag와 end tag를 연결 리스트의 구조체로 설정한 이유는 한 단어에 하나의 태그만이 붙는 경우는 매우 드물기 때문이다. 또한 head와 tail의 이중 연결 리스트를 사용하여 태그의 마지막을 확인하기 쉽도록 한다.

전처리기에서는 고유명사와 날짜를 인식하면 해당 단어나 구에 대해서 자질값을 부여한다. 이 자질값은 영한 기계번역기에서 유용하게 사용할 수 있다. 이 자질과 자질값을 저장하기 위한 변수로 f_name과 f_value를 사용한다.

단어와 태그의 분리는 태그의 시작을 나타내는 문자인 '<'를 이용하며 start tag와 end tag의 분리는 태그의 두 번째 문자인 '/'가 있는지를 비교하여 구분한다. end tag를 기준으로 현재 처리해야 할 단어나 태그를 기존의 노드에 추가할 것인지 새로운 노드를 만들 것인지를 결정한다.

5. 구현 및 실험

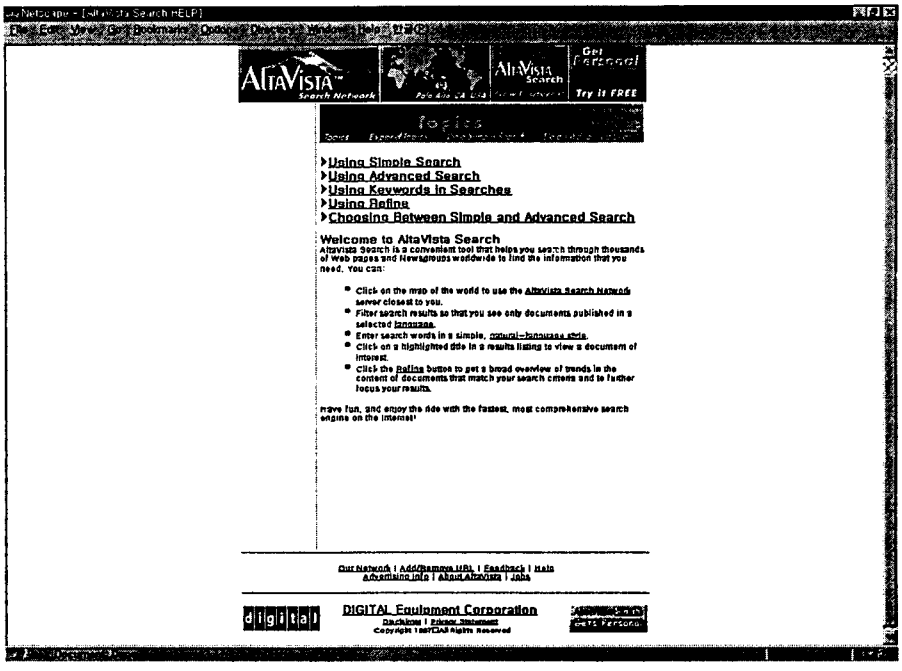
본 논문에서 제안한 문서 전처리기는 lex[John 93]를 이용하여 C 언어로 구현하였다. lex를 이용한 전처리기는 프로그램의 유지 보수가 매우 용이하고 새로운 현상에 쉽게 대처할 수 있다.

<그림 3>은 번역 대상 웹 문서이다. AltaVista의 Help 문서이다.

<그림 4>는 문서 전처리기가 번역 대상 웹 문서에 대해 처리를 수행한 후의 결과이다. 번역 대상 웹 문서의 일부분만을 보여주고 있다. 문장 단위로 구분이 되어 있으며 단어와 태그를 구분할 수 있다. 단어가 아닌 태그에는 자질값이 부여되어 있어서 쉽게 후처리를 할 수 있다.

<그림 5>는 영한 기계번역기가 문서 전처리기의 결과를 이용할 수 있도록 문장, 자질, 태그를 분리한 결과이다. 각 단어에는 한 문서에서의 문장 번호와 각 문장 내에서의 단어 번호가 붙어 있다.

본 시스템의 수행 결과를 평가해 보면 하이픈 처리, 고유명사 인식, 특수기호 처리, 대소문자 정규화, 날짜 처리, HTML 태그 분리 등이 매우 성공적이다.



<그림 3> 번역 대상 웹 문서

```
;; Sent 10
  (("<br>" :TAG HTML) ("<IMG SRC=\"images/arrow_dkblue.gif\" HEIGHT=14 WIDTH=10>" :TAG HTML) ("<a
href=\"help_general.htm\">" :TAG HTML) ("choosing") ("between") ("simple") ("and") ("advanced") ("Search")
("</a>") )
;; Sent 11
  (("</b>" :TAG HTML) ("<p>" :TAG HTML) ("<A NAME=\"intro\">" :TAG HTML) ("</A>") )
;; Sent 12
  (("<B>" :TAG HTML) ("welcome") ("to") ("AltaVista") ("Search") ("</B>" :TAG HTML) ("</FONT>" :TAG
HTML) ("<BR>" :TAG HTML) ("<FONT FACE=\"arial, helvetica\" size=\"-1\">" :TAG HTML) ("AltaVista")
("search") ("is") ("a") ("convenient") ("tool") ("that") ("helps") ("you") ("search") ("through") ("thousands")
("of") ("Web") ("pages") ("and") ("newsgroups") ("worldwide") ("to") ("find") ("the") ("information") ("that")
("you") ("need") (".") )
;; Sent 13
  (("you") ("can") (":") ("</FONT>" :TAG HTML) ("<UL>" :TAG HTML) ("<LI>" :TAG HTML) ("<FONT
FACE=\"arial, helvetica\" size=\"-1\">" :TAG HTML) ("click") ("on") ("the") ("map") ("of") ("the") ("world")
("to") ("use") ("the") ("<A HREF=\"av_network.htm\">" :TAG HTML) ("AltaVista") ("search") ("Network")
("</A>") )
;; Sent 14
  ( ("server") ("closest") ("to") ("you") (".") ("</FONT>" :TAG HTML) ("</LI>" :TAG HTML) ("<LI>" :TAG
HTML) ("<FONT FACE=\"arial, helvetica\" size=\"-1\">" :TAG HTML) ("filter") ("search") ("results") ("so")
("that") ("you") ("see") ("only") ("documents") ("published") ("in") ("a") ("selected") ("<A
HREF=\"help_general_languages.htm\">" :TAG HTML) ("language") ("</A>") )
;; Sent 15
  (("." ) ("</FONT>" :TAG HTML) ("</LI>" :TAG HTML) ("<LI>" :TAG HTML) ("<FONT FACE=\"arial,
helvetica\" size=\"-1\">" :TAG HTML) ("enter") ("search") ("words") ("in") ("a") ("simple") (",") ("<A
HREF=\"help_simple.htm\">" :TAG HTML) ("natural-language") ("style") ("</A>") )
```

<그림 4> 문서 전처리기 수행 결과 (일부분)

```
 10,     1, choosing
start tag: <br>
start tag: <IMG SRC="images/arrow_dkblue.gif"
           HEIGHT=14 WIDTH=10>
start tag: <a href="help_general.htm">
 10,     2, between
 10,     3, simple
 10,     4, and
 10,     5, advanced
 10,     6, Search
  end tag: </a>
 11,     1, , NULL
  end tag: </b>
 11,     2, , NULL
start tag: <p>
start tag: <A NAME="intro">
  end tag: </A>
 12,     1, welcome
start tag: <B>
 12,     2, to
 12,     3, AltaVista
 12,     4, Search
  end tag: </B>
  end tag: </FONT>
 12,     5, AltaVista
start tag: <BR>
start tag: <FONT FACE="arial, helvetica" size="-1">
 12,     6, search
 12,     7, is
 12,     8, a,
 12,     9, convenient
 12,    10, tool
 12,    11, that
 12,    12, helps
 12,    13, you
 12,    14, search
 12,    15, through
 12,    16, thousands
 12,    17, of
 12,    18, Web
 12,    19, pages
 12,    20, and
 12,    21, newsgroups
 12,    22, worldwide
 12,    23, to
 12,    24, find
 12,    25, the
 12,    26, information
 12,    27, that
 12,    28, you
 12,    29, need
 12,    30, .,
```

<그림 5> 문장, 태그, 자질 분리 결과 (일부분)

본 시스템은 아직 문장 분리 기능이 아직 완전하지 못하다. 문장 중간에 나오는 다른 웹 문서로 연결해 주는 링크가 독립된 문장으로 분리되고 있다. 독립적인 링크와의 구분이 단순히 태그만을 가지고는 처리되지 않는다. 또한, 고유명사의 향상된 인식을 위해서는 고유명사 사전의 보강이 필요하다.

웹 문서 저작도구의 사용으로 웹 문서에 HTML 태그가 매우 많아졌다. 이 영한 기계번역기에서 번역을 하기 위해 필요하지 않은 양의 태그 세트를 단어와 함께 계속 가지고 다니는 것은 매우 부담이 되고 번역 수행에 별다른 도움이 되지 않는다. 따라서, HTML 태그는 파일에 저장을 하고 해당 태그 세트 번호만을 가지고 가도록 한다.

6. 결론

본 연구에서는 영어 웹 문서를 한국어로 기계번역을 수행하기 위하여 문장 인식 및 분리, 타이틀 처리, 나열된 단어의 처리, 하이픈 처리, 특수기호 처리, 대소문자 정규화, 고유명사 인식, 날짜 인식, HTML tag를 처리하는 웹용 영한 기계번역을 위한 문서 전처리기를 구현하였다.

전처리기는 웹 문서에서 HTML tag를 분리하여 번역 대상 문장을 구성하며 복합 단위를 인식하고 고유명사와 날짜의 자질값을 부여하여 영어 형태소 분석기의 처리 부담을 줄이고 성능을 향상시킨다.

또한, 다양한 정규 표현과 문법 규칙을 제공하는 lex를 이용하여 전처리기를 구현함으로써 새로운 현상을 쉽게 반영할 수 있고 프로그램의 유지 보수가 용이하며 처리속도가 매우 빠르다.

수행 결과를 평가해 본 결과 HTML 태그의 분리, 하이픈 처리, 대소문자의 정규화 등의 대부분의 기능이 우수하였으나 문장 분리 기능의 성능이 미흡하였다.

참고 문헌

[Collins 90] Collins Cobuild English Grammar, Collins, 1990

[John 93] John R. Levine, Tony Mason & Doug Brown, lex & Yacc, O'Reilly & Associates, Inc, 1993

[Penn 97] http://linc2.cis.upenn.edu/~treebank/home.html

[여상화 95] 여상화, 정한민, 채영숙, 김태완, 박동인, "실용적인 영한 기계번역을 위한 전처리기의 설계 및 구현," 1996년도 제8회 한글 및 한국어 정보처리 학술대회, 1996, pp.313-319

[KAIST 92] 한국과학기술원, 영한기계번역시스템(III): 문법개발지원환경 및 해석문법개발, 과학기술처, 1992